UNIVERSITAT POLITÈCNICA DE CATALUNYA

Brain lesion segmentation using Convolutional Neuronal Networks

by

Clara Bonnín Rosselló

In partial fulfilment of the requirements for the degree in
Ciències i Tecnologies de Telecomunicació engineering

in the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Image Processing Group

Advisors: Verónica Vilaplana and Adrià Casamitjana

May 2018

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Abstract

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Image Processing Group

by Clara Bonnín Rosselló

Convolutional neural networks (CNN) are powerful tools for learning representations from images. They are being used in a large range of applications, being the state-of-the art in many computer vision tasks. In this work, we study the brain tumor segmentation problem using CNNs and the publicly available BraTS dataset. One of the key factors for this task is which training scheme is used since it should deal with memory constraints and should alleviate the high-imbalance nature between healthy and lesion tissue in the brain.

Thus, the purpose of this project is to propose a comparison between several training schemes and extensively analyze and evaluate them in terms of the dice score. We evaluate dense-training against patch-sampling, and particularly, fixed-rule against adaptive sampling scheme. Furthermore, variants and modifications of the existing training schemes have been proposed in order to enhance their performance. Finally, several loss functions for each training scheme have been analyzed.

Acknowledgements

I would like to express my sincere gratitude to several people. Firstly, I would like to thank specially Adrià Casamitjana and Verónica Vilaplana, the supervisors of this thesis, for their advice and patience during the whole project and for giving me the chance to learn deep learning in this challenging work that combines biomedicine and engineering. Of course, I would like to thank for the collaboration scholarship received to realize this thesis. I would like to take this opportunity to thank the ETSETB teachers and also my colleges for this last wonderful 4 years in the UPC. Finally, I would like to dedicate this project to my family and to Òscar, for their support and understanding.

Contents

List of Figures

List of Tables

Abbreviations

BaseASS	**B**aseline **A**daptive **S**ampling **S**cheme
CNN	**C**onvolutional **N**eural **N**etwork
CASED	**C**urriculum **A**daptive **S**ampling for **E**xtreme **D**ata Imbalance
DSC	**D**ice **S**imilarity **C**oefficient
FLAIR	**F**luid **A**ttenuated **I**nversion **R**ecovery
HGG	**H**igh **G**rade **G**lioma
ROI	**R**egion **O**f **I**nterest
LGG	**L**ow **G**rade **G**lioma
MRI	**M**agnetic **R**esonance **I**maging
WL	**W**eighted **L**oss

Symbols

N	Number of classes
G	Ground Truth
P	Prediction
y	True Labels
$\hat{y}$	Softmax output values
$\mathcal{L}$	Loss
$\mathcal{L}_w$	Weighted Loss
$\mathcal{H}$	Hausdorff Distance
TP	True positive
FP	False positive
E_i	Error Map of the i-th training image
w	Training weights

I would like to dedicate this work to my grandparents for being such excellent people, and especially to my grandmother who died of cancer.

Chapter 1

Introduction

Glioma is the most common brain tumor family, which rises from glial cells and invades the enclosing tissues [1]. Patients with the hardest and more aggressive variant of this tumor (high-grade gliomas) have a life expectancy of two years or less under strict treatment. Neuroimaging protocols are necessary throughout all the course of the disease in order to evaluate the illness' progression and measure the success of a certain treatment [2].

1.1 Statement of Purpose

Manual tumor segmentation is a struggling task and needs to be done by an experienced specialist, while imaging processing algorithms can automatically analyze many brain tumor scans in far less time. Automatic segmentation has as a huge potential to improve diagnosis, treatment election and tracking [2]. However, computerized brain tumor segmentation is a challenging task since tumor structures are different in each patient in terms of size, location and shape.

Multimodal magnetic resonance imaging (MRI) is the principal method of screening and diagnosis for brain tumor. The lesion is identified through the relative intensity changes in comparison to the baseline tones of the healthy tissue. In this work, BraTS'17 dataset [3] has been used: it includes data acquired for four different MRI modalities and the ground truth labels which have been manually checked by certified neuroradiologists. The multimodal scans considered in the project are: T1, T1c, T2 and FLAIR. In figure 1.5, it can be appreciated the four different brain scans for the same subject.

The purpose of this work is to explore state-of-the-art deep learning techniques for image segments from 3D images and provide a system that achieves good results on automatic brain tumor segmentation. The segmentation task consists on classifying at the smallest addressable scale (voxel unit) the MRI image as background or one of the three tumor's subregions: necrotic core (class 1), edema (class 2) and enhancing tumor (class 3). See figure 1.6.

Convolutional neural networks (CNN) have been chosen because of their good performance when working with images and with three-dimensional inputs. The CNN architecture makes the

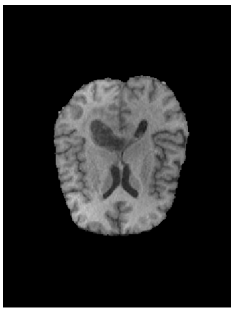


FIGURE 1.1: MRI T1

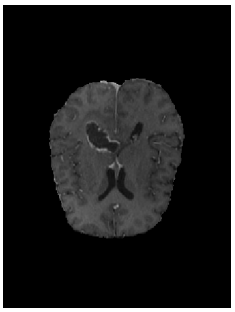


FIGURE 1.2: MRI T1c

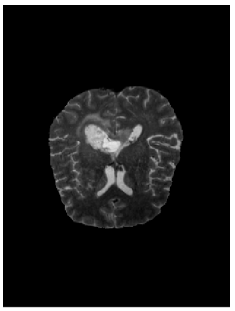


FIGURE 1.3: MRI T2

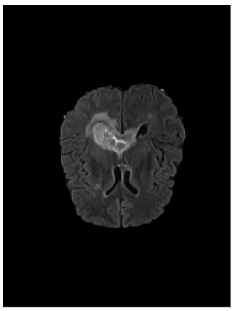


FIGURE 1.4: MRI FLAIR

FIGURE 1.5: Multimodal MRI images from subject $Brats17 - CBICA - ALX$

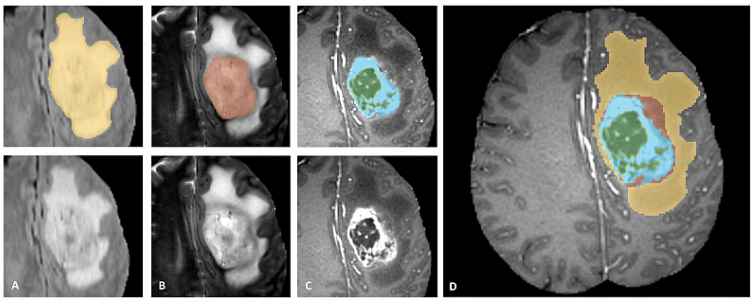


FIGURE 1.6: Glioma sub-region. Shown are image patches with the tumor sub-regions that are annotated in the different modalities (top left) and the final labels for the whole dataset (right). The image patches show from left to right: the whole tumor (yellow) visible in T2-FLAIR (Fig.A), the tumor core (red) visible in T2 (Fig.B), the enhancing tumor structures (light blue) visible in T1Gd, surrounding the necrotic components of the core (green) (Fig. C). The segmentations are combined to generate the final labels of the tumor sub-regions (Fig.D): edema (yellow), non-enhancing solid core (red) and enhancing tumor formed by the necrotic core (green) and enhancing core (light blue). Figure and annotation taken from [4]

forward function more efficient and vastly diminishes the number of parameters in comparison to regular neural networks [5].

The use of convolutional neural networks in medical images, and particularly in brain tumor segmentation, arises many problems. Firstly, the localization of gliomas and glioblastamas is difficult as they do not have easy and clearly defined borders. Another issue is the high imbalance between background voxels and different tumor regions, as tumor represents a far smaller area of MRI volumes. Particularly, in our dataset, the healthy tissue comprises 98.2% of the total voxels and the remaining 1.8% is distributed among the pathology subregions: 0.3 % belongs to necrosis and non-enhancing tumor, 1.1% edema and 0.4 % to enhancing-tumor [6]. The natural true distribution overwhelmes the network such that a naive training scheme would provide a model predicting erroneously all tissue as healthy. Hence, the network must be tricked in order to achieve good classification results. In this work, we provide several solutions to these challenging problems. Two methods are proposed: (1) Modification of the loss function to emphasize minoritary classes and (2) the smartly selection of input data to modify the training voxel distribution. This last procedure can be done by dividing the original image in smaller segments and it is called patch sampling.

1.2 Outline of the work

An introduction on automatic brain tumor segmentation has been done in this section and a brief analysis on the state-of-the-art will be presented in Chapter 2.

Chapter 3 accounts for the methodology and theoretical background necessary to correctly appreciate the experimental section. The different procedures and experiments are developed and discussed in Chapter 4. The budget of the project will be detailed in Chapter 5. The conclusions drawn from the Chapter 4 will be discussed in Chapter 6, as well as possible ideas for future work. And lastly, it will be found the Appendix where additional information on the reported results will be attached.

1.3 Technical Remarks

The project was not started from scratch, the core code used was presented in [7]. The project has been developed in Python using Keras [8] framework. Keras is a high-level neural network API capable of running on top of Tensorflow.

Also it has to be mentioned that the software FSL-Eyes was used in order to visualize the MRI original images and the predictions done. And finally, in addition to the software a GPU was required due to the high demanding computational effort to train convolutional neural networks.

Chapter 2

State of the art

The rise of deep learning for computer vision and, particularly, for semantic segmentation tasks makes attractive its use for medical image segmentation. Convolutional Neural Networks (CNNs) have been applied with promising results on different medical imaging problems [2].

Convolutional Neural Networks are very similar to ordinary Neural Networks: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and follows it with a non-linearity (sigmoid, ReLu...). The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function on the last layer. However, CNN architectures make the assumption that the inputs are images, which allows to encode certain properties into the architecture (local connectivity, parameter sharing and invariance to local changes through pooling operations) [9].

In the past years, medical image segmentation and specially brain tumor segmentation, has moved towards deep learning solutions using CNNs. Initially, two-parallel-path architectures were proposed (Pereira et al. [10], Havaei et al. [6]; Kamnitsas et al.[11]) providing good performance. These approaches exploited both local features as well as more global contextual features simultaneously. Other schemes, as the encoder-decoder have also been succesful [12]. U-net 3D [13] extends the previous u-net architecture from Ronneberger et al. [14] by replacing all 2D operations with their 3D counterparts. Highway net ([15]) allows unimpeded information flow across several layers on information highways and uses gating units which learn to regulate the flow of information through a network.

Recently, other schemes without pooling have shown good performance, improving in some cases the detection and tumor delineation. HighResNet [16] is a high-resolution and compact network architecture for the segmentation of fine structures in volumetric images and introduces new elements as dilated convolutions and residual connections. Finally, at the last BraTS challenge [17] an ensemble of different architectures has been proposed in [18], searching to improve the result of each one separately.

But not just the architecture election is the key element. Other problems, as the available memory, gradient flux or normalitzation are active topics in this research field. This encourages

the emergence of different sampling schemes (dense-training, patchwise training, etc..) and cost functions.

Performance of CNN is significantly influenced by the strategy of extracting training samples. A common approach is selecting image patches equally sampled from each class. Another approach is to equally sample background and foreground segments. On the other hand, by employing dense-training or by sampling patches uniformly, it might suffer from severe class imbalance. Hence, multiple cost functions have been proposed to alleviate this issue.

The loss function proposed is cross-entropy. However, when the training data is severely unbalanced, this formulation can lead to a strongly biased estimation towards the majority class. A weighted cross-entropy (Brosch et al. [19]) is proposed to tackle this problem, where the weights are inversely propotional to the class frequencies. Also, a differentiable version of the Dice Score Coefficient (DSC), proposed by Milletari et al. [20], is used as loss function as it measures the overlap of the region of interest (ROI). Recently, two novel loss functions have been presented: the Generalised DSC [21] and the Wasserstein Dice .

This work focuses on the segmentation of brain tumors following the guidelines indicated by BraTS Challenge, which began in 2012. The methods submitted in these last years can be found in [17], [22]. The MICCAI (The Medical Image Computing and Computer Assisted Intervention Society) BraTS Challenge has an updated leaderboard[1] of the models with the best performances according to following metrics: Dice score, sensitivity, specificity and Hausdorff distance. UCL-TIG is in the first position on the ranking with *Ensembles of Multiple Models and Architectures for Robust Brain Tumour Segmentation* [18]. It achieves a dice whole tumor of 0.90499, a dice core tumor of 0.83779 and and a dice enhancing tumor of 0.78585.

[1] https://www.cbica.upenn.edu/BraTS17/lboardValidation.html

Chapter 3

Methodology

3.1 System Architecture

3.1.1 Architecture

Two different architectures have been studied in order to discern which one combined with other configurations has the best behavior: The Masked V-Net [12] and the Deep Medic [11].

3.1.1.1 Masked V-NET

The masked V-Net [12] is a modified version of the V-Net architecture [20], which consists of a downsampling or encoder path in charge of compacting the signal and an upsampling or decoder path that combines coarse features from the encoder output with fine features from hidden, intermediate levels of the encoder to provide a segmented image of the same resolution as the input image. The modifications include the use of small kernels of size 3^3, batch normalization after the convolution and then ReLU as non-linearity. It was also introduced a modified expression for the residual connections that aim to preserve the input signal through all the network. Max-pooling, repeated up-sampling for spatial correspondence and 1x1x1 convolution are variants introduced to ensure dimensions matched in the addition layer. Note that the ROI mask introduced before the final predictions was only used for the dense-training experiments. See figure 3.1.

3.1.1.2 Deep Medic Network

The Deep Medic architecture proposed in [11] consists in capturing contextual and spacial information through two parallel paths with different feature resolution that saves computational costs and avoids *pooling* which could affect on the accuracy of our system. The 11-layer architecture proposed by *Deep Medic* [11] is built as shown in figure 3.2. A high resolution path is able to capture the most complex details within a small local neighbourhood, while a parallel low resolution path captures image-level features such as localization or tumor size. The difference in

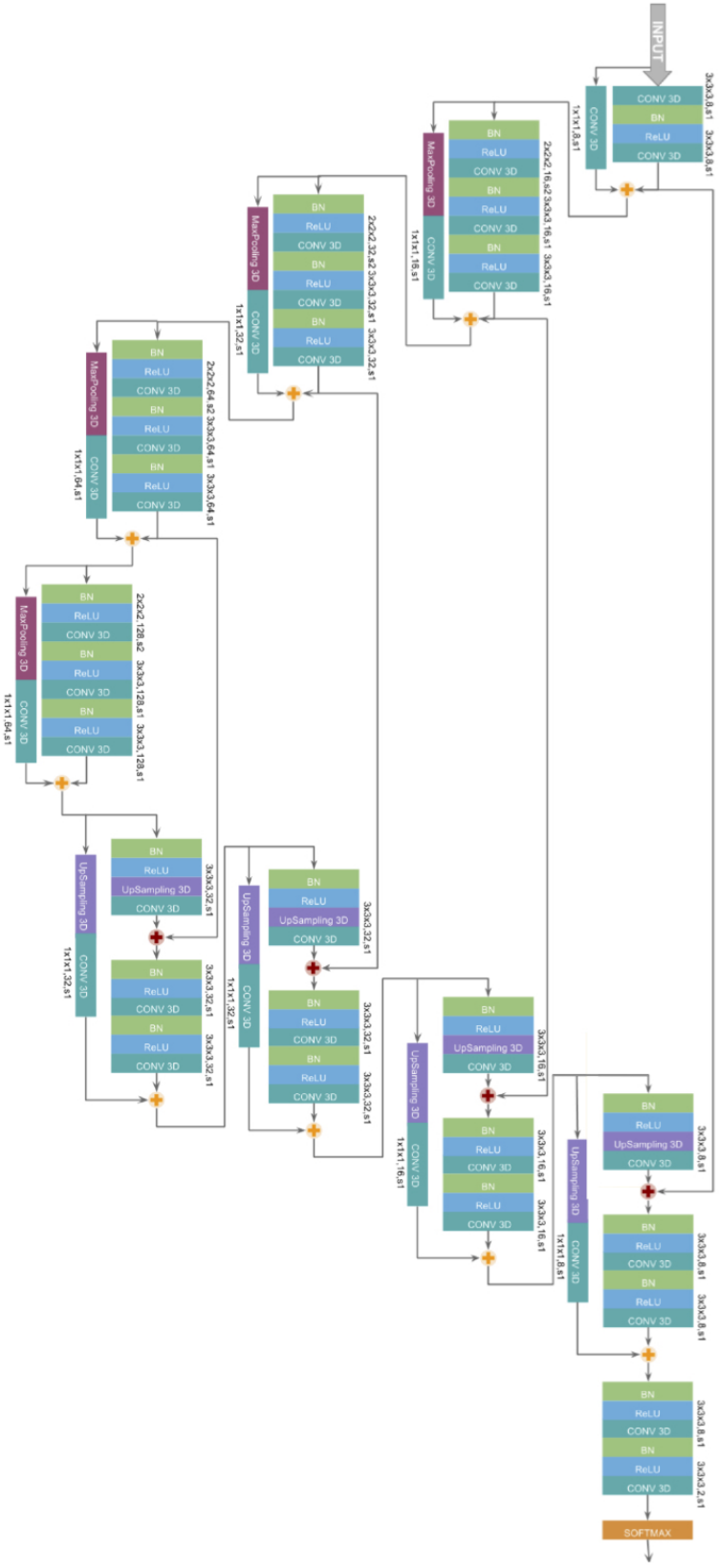


FIGURE 3.1: V-Net

resolution is achieved with different receptive fields: both paths are built upon a concatenation of convolutional layers but the later has a pooling module at the very beginning.

The kernels on the convolutional layers of both high and low resolution paths are of size 3^3. The resulting matrices of the convolutional layers are first combined into two full classification layers and then finally classified.

3.1.2 Loss functions

Different loss functions have been considered in order to study which one fits better to our segmentation problem. The three cost functions that will be studied are: cross-entropy, Dice Similarity Coefficient (DSC) and Generalised DSC.

3.1.2.1 Cross-entropy

Cross-entropy loss [23] for a multi-class setting can be expressed as follows

$$\mathcal{L}(\hat{y}, y) = -\sum_{n=1}^{N} y_n log(\hat{y_n}) \tag{3.1}$$

Note that N stands for the number of classes and y and $\hat{y}$ are N-dimensional vectors, where y_n are the true labels for the class n and $\hat{y_n}$ are the softmax outputs of the network for those true labels. The mean cross-entropy over the whole batch is used as the cost function at each iteration and is computed as follows:

$$\bar{\mathcal{L}}(\hat{y}, y) = -\frac{1}{|Y|}\sum_{y}\sum_{n} y_n log(\hat{y_n}) \tag{3.2}$$

where Y refers to training samples from the batch.

3.1.2.2 Dice Similarity Coefficient

Dice Score for multi-class segmentation [24] is a measure of similarity between two binary sets: the ground truth G and the prediction P. Each set consists of a region of interest (ROI) and background and the dice score, $\mathcal{D}\epsilon[0, 1]$, is the ratio between the intersection of the ground-truth and prediction ROIs and the sum of the areas of both ROIs. The following expression explains the idea behind this loss function.

$$\mathcal{D} = \frac{2\,|P \cap G|}{|G| + |P|} \tag{3.3}$$

Then, a continuous and differential approximation can be done by using softmax predictions instead of the predictions themselves. The result of averaging and adapting the previous expression

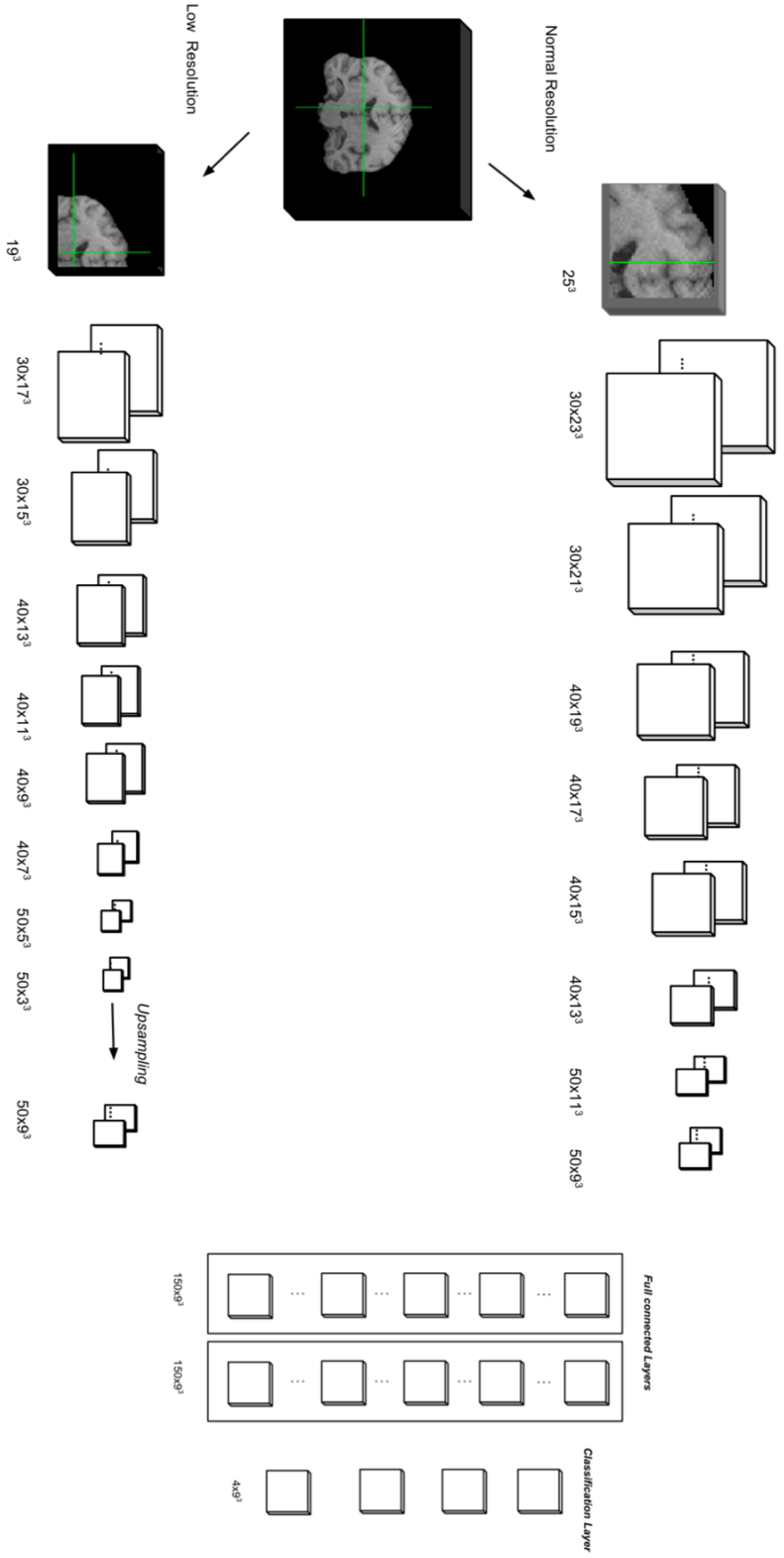


FIGURE 3.2: Two-path Deep Medic architecture

to our multi-class problem is as follows

$$\bar{\mathcal{L}}(\hat{y}, y) = \frac{1}{|N|} \sum_{n \epsilon N} \frac{2 \sum_i y_n^i \hat{y_n^i}}{\sum_i (y_n^i + \hat{y_n^i})} \tag{3.4}$$

3.1.2.3 Generalised Dice Score

The generalised Dice Score is proposed as loss function in [21]. Its a modified version of the DSC and is given by:

$$\bar{\mathcal{L}}(\hat{y}, y) = 1 - 2 \frac{\sum_{n=1}^{N} \alpha_n \sum_i y_n^i \hat{y_n^i}}{\sum_{n=1}^{N} \alpha_n \sum_i (y_n^i + \hat{y_n^i})} \tag{3.5}$$

where α_n is a weight to balance the impact in the loss function of class n. Weighting by the inverse of the class' volume corrects the contribution of each label and reduces the correlation between the region size and the Dice score. It is calculated as the inverse of the squared sum of all the voxels of class n:

$$\alpha_n = \frac{1}{(\sum_i y_n^i)^2}$$

3.1.2.4 Weighted loss

To further eliminate the negative impact of the class imbalance, a weighted loss $\mathcal{L}_w$ is proposed as follows. Where $\mathcal{L}_n$ is the specific loss for a certain class $n \epsilon N$ and $\|\alpha_n\|$ is the the probability of appearance of that certain class. Therefore, by inverting this probability, we manage to give more weight to classes that appear much less frequently than others.

$$\mathcal{L}_w = \sum_{n \epsilon N} \frac{1}{\alpha_n} \mathcal{L}_n \tag{3.6}$$

The class frequency α_n is calculated as the sum of all the voxels of class n divided by the sum of all the voxels of the N classes:

$$\alpha_n = \frac{\sum_i y_n^i}{\sum_{n=1}^{N} \sum_i y_n^i} \tag{3.7}$$

3.1.3 Metrics

The metrics used to evaluate the performance of each of the experiments carried on are detailed below.

3.1.3.1 Dice Score

The evaluation of the method using the dice score was calculated according to 3.4 for different ROI definitions: tumor core (classes 1 and 3), whole tumor (classes 1,2,3) and enhancing tumor (class 3). This evaluation framework is imposed by the Multimodal Brain Tumor Segmentation Challenge [3].

3.1.3.2 Confusion matrix

The confusion matrix is a table which allows to evaluate if our segmentation system is mislabelling one class as another. Each row of the table represents the true class and each column the predicted class. This table is useful to give an idea of which classes are well classified and which are wrongly confused with the others. It gives us insights of how a model can be improved. The confusion matrices in this document will present the following structure:

G \ P	**0**	**1**	**2**	**3**
0				
1				
2				
3				

TABLE 3.1: Confusion Matrix Example

3.2 Training scheme

3.2.1 Dense-training

Before analyzing how we explore the patch wise training scheme, which is the main contribution of our work, we study dense-training. We denominate dense-training the strategy that uses the whole MRI image and the four modalities as input into the our Convolutional Neural Network. The performance of this training scheme will be analyzed for two different loss functions.

3.2.2 Patch sampling

Patch-sampling training scheme consists of feeding the network with small three-dimensional patches of each subject. Each of these slices is associated with 4 different modalities: T1, T2, T1C and FLAIR and its size is set to 64^3. The sampling scheme can be critique for any medical application and an exhaustive analysis of different methods is performed throughout the manuscript with further numerical comparison. In the case of brain tumor segmentation, the high imbalance between background and tumor regions and subregions may require flexible method that balances the input training distribution of the different classes.

3.2.2.1 Baseline: Foreground-background

The training scheme used in [11] tries to solve the imbalance problem by a sampling scheme that samples the central-voxel of each patch with equiprobability between foreground (tumor regions) and background. Hence, each batch is build by the same number of patches whose central voxel is foreground and background. Note that no distinction is made between tumor subregions. This is to maintain the relative distribution of the foreground classes and at the same time account for the imbalance problem between healthy tissue and tumor tissue.

Figure 3.3 from [11] shows how the relative distribution of the foreground classes is closely preserved and the imbalance in comparison to the healthy tissue is automatically alleviated.

	Healthy	Necrosis	Edema	Non-Enh.	Enh.Core
Real	92.42	0.43	4.87	1.02	1.27
Captured	58.65	2.48	24.98	6.40	7.48

FIGURE 3.3: Example of real vs captured distribution in the training data of BRATS 2015

This foreground-background sampling scheme has been selected to be our baseline scheme on patch sampling.

3.2.2.2 Per-Class sampling scheme

This training scheme proposes sampling patches according to a rule that ensures equiprobability between all classes in the central-voxel at each epoch. The idea is to account for the imbalance between background and each one of the tumor subregions. It alters the balance between each class and hence, it captures a rather different distribution from the original.

3.2.2.3 Curriculum Adaptive Sampling

Curriculum Adaptive Sampling (CASED) [25] scheme first's objective is to tackle the problem of class imbalance. The basic ideas of this system are:

1. Learn features related to tumor: start introducing only patches with tumor into the network.
2. As the network is training, start adding background patches to learn healthy tissue properties.
3. In the end, uniform sampling is reached to mimic real data distribution.

CASED scheme can be divided in two parts: **Curriculum** and **Adaptive** Sampling.

Curriculum is the part that tackles the class imbalance problem controlling the input patches to the network by deciding between tumor-sampling and uniform-sampling generators. If training was performed using only tumor pathces, it could result in overfitting because it would not learn how to represent the main part of the MRI image, the background class. Then, the curriculum part is responsible of decreasing (as function of the training examples seen) the number of patches with tumor until it reaches the real distribution. The threshold p_x is given by Equation 3.8. For values greater than p_x tumor patches are selected, otherwise it is picked any random patch.

$$p_{x+1} = p_x * \frac{1}{M}^{(iter*epochs*K)^{-1}} \tag{3.8}$$

with

$$p_0 = 1$$

where M is the number of segments, *iter* the number of iterations, *epochs* the total number of training epochs and K is a constant to speed up or slow down the threshold curve.

Adaptive Sampling is required to refine the previous part. Even using the curriculum, solutions with false positives would still happen. Moreover, mostly all voxels on brain images could be confidently classified as background. Therefore, the adaptive sampling encourages training inputs whose predictions are false positives.

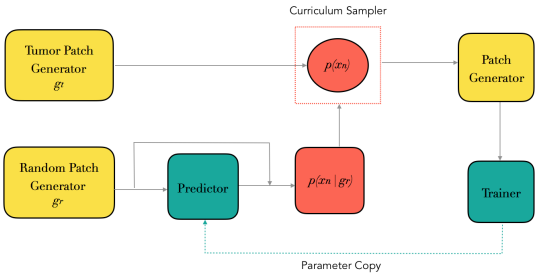


FIGURE 3.4: Schematic diagram of CASED framework

3.2.2.4 Baseline Adaptive Sampling Scheme

Adaptive Sampling Scheme to Efficiently Train Fully Convolutional Networks [26], from now on BaseASS (baseline adaptive sampling scheme), suggests a patchwise training scheme that pretends to adaptively build traning samples at each epoch by looking at the network training error. For each subject, error maps E_i are build concurrently at the end of each epoch as:

$$E_i = 1 - CNN(w, I_n(x))_{L_n(x)} \tag{3.9}$$

where $CNN(w, I_n(x))_{L_n(x)}$ represents the softmax predictions calculated using the training weights w over an image I_n. Thus, a patch is accepted into the batch according to its relation to the threshold defined by:

$$E_i(c) > U(0,1) - \epsilon \tag{3.10}$$

where c is the central voxel of the patch, $U(0,1)$ is a random uniform variable and ϵ a parameter to calibrate the algorithm: $\epsilon = 0$ means a completely adaptive scheme and $\epsilon = 1$ the uniform sampling scheme.

Chapter 4

Experiments and Results

This chapter presents and compares the results obtained from applying the methodologies mentioned in Chapter 3.

4.1 Dataset

In this thesis, the data used to train the network has been obtained from the MICCAI BraTS 2017 Challenge [3]. This dataset is composed by 210 HGG (high grade glioma) and 75 LGG (low grade glioma) subjects of which 171 are used for training (60%) and 116 are used for validation purposes (40%). No data augmentation was used in any experiment. The four modalities of each MRI image (native T1, post-contrast T1-weighted, T2-weighted and T2 FLAIR) are co-registered to the same anatomical template and interpolated to the same resolution (1 mm^3).

Training the network relies on the proper selection of hyperparameters, a wrong choice can lead to overfitting, underfitting or simply not training. Each experiment should have been optimized individually until obtaining the best results. Besides the fact that this is too computationally demanding, we have not done this in order to be able to compare all the models under the same conditions. First, the learning rate, which tells the optimizer how far has to move the weights in the direction of the gradient, was set to 0.0005. The momentum was set to 0.99. Regularization prevents the coefficients to overfit and it depends on two variables: L2 is a factor multiplying the sum of the square of the weights, while L1 is the factor multiplying the absolute sum of the weights. The values chosen were L1=0.00001, L2=0.005. The number of training epochs is variable for each experiment. Masked V-Net was used as the base architecture for all experiments except one, where Deep Medic Network was used. The initialization of the weights was done according to [27].

For patch sampling, we used 600 segments/epoch for training and 400 segments/epoch for validation with a batch size of 10.

4.2 Dense-training

We begin exploring the dense-training scheme vastly used in natural 2-D images, in which CNN are trained using the whole-subject. The network processes 171 training images of size $[192, 192, 160]$ each epoch using the Adam optimization method [28]. In each iteration CNN's parameters (weights and biases) are updated in order minimize the cost function. In the context of this work, we are going to analyze the impact on the segmentation performance of two different loss functions as a way to balance the distribution of the training samples.

The different loss functions used in the experiment are compared in Table 4.1. We observe that training this network with cross entropy loss function (Eq. 3.2) leads to poor segmentation results. Instead, training with Dice loss function (Eq. 3.4) appears to be more robust to class imbalance problem. This result makes sense as cross-entropy is more sensitive to the balance of classes because it tries to minimize, in mean, the agreement between classes, whether they are tumour or background, and therefore does not distinguish between true-positives and true-negatives. Concerning the DSC, it takes more into account the positive class as it gives more weight to true-positives than true-negatives.

Loss Function	**Dice Whole**	**Dice Core**	**Dice Enhance**
Cross-entropy	0,67032	0,41302	0,50329
Dice similarity coefficient	0,80534	0,70092	0,65296

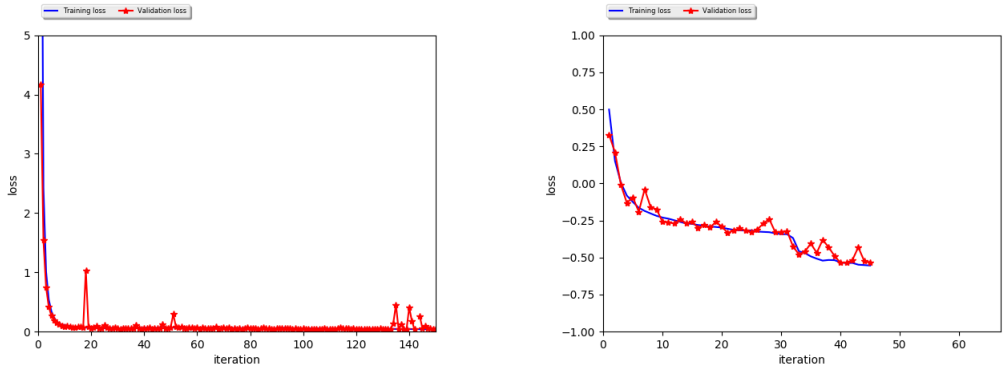
TABLE 4.1: Comparison of mean validation DSC metrics for dense-training

Figure 4.1 compares the convergence of the loss functions studied. The blue curve indicates the training loss and the red curve the validation loss. As seen in figure 4.1 (a), the cross-entropy is minimized in a vecinity close to its optimal value zero. The dice coefficient loss function converges to a non-optimal value (far from -1) and it is slower than cross-entropy function, as shown in Figure 4.1 (b). This is an interesting result because although cross-entropy reaches the optimum value and DSC does not, the latter achieves much better inference results (Table 4.1). This fact indicates that the second loss is much better tailored (for this particular configuration) to our segmentation task.

G \ P %	**0**	**1**	**2**	**3**
0	**0,99890**	0,00005	0,00085	0,00020
1	0,23088	**0,46246**	0,25662	0,05004
2	0,21386	0,02957	**0,73939**	0,01718
3	0,05634	0,06851	0,04991	**0,82523**

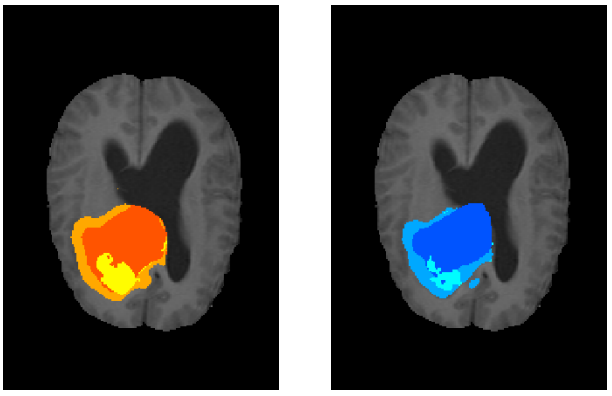
TABLE 4.2: Confusion matrix for whole subjet using DSC cost function (validation)

Finally, we obtained the confusion matrix for dense-training using DSC as cost function (Table 4.2) where the number of false positives for class 0 is considerable high, damaging class 1 and 2. This can also been demonstrated by qualitative results, shown in Figure 4.2 (training with DSC). It can be seen that this method correctly separates background (class 0) from tumor (class 1,2,3). However, the segmentation of the different tumor subregions gives poor results. This results motivates us to explore other solutions.



(a) Cross-entropy (b) Dice similarity coefficient

FIGURE 4.1: Loss function convergence for dense-training



(a) Ground Truth (b) Prediction

FIGURE 4.2: Ground Truth vs Prediction for subject $Brats17-TCIA-444-1$. In (a) Ground Truth: dark orange (class 1), light orange (class 2) and yellow (Class 3). In (b) Prediction: dark blue (class 1), light blue (class 2) and cyan (class 3)

4.3 Patch sampling

After analyzing how our network performs when using the whole subject as training input, we want to see how well patch sampling helps in the segmentation of brain tumors. In this section we will observe the segmentation performance of different experiments. First, we try to deal with class imbalance choosing small patches of size 64^3 and sampling with two different fixed-rule sampling strategies: Foreground-background [11] sampling scheme and Per-class sampling scheme. Second, we implement two different adaptive sampling strategies: BaseASS [26] and CASED [25].

4.3.1 Fixed-rule sampling schemes

In this experiment, we primarily want to to compare both fixed-rule sampling strategies. We want to see which variant performs better in detecting the tumor borders and the different subregions. The foreground-background sampling strategy modifies the tumor/non-tumor distribution while

preserves the relation among the tumor's subregions. This is done by sampling 50% background (Class 0) and 50% foreground (tumor classes 1,2,3). Per-class sampling scheme consists in sampling equiprobably from all four classes (% 25 for class n with $n = 0, 1, 2, 3$) so that the networks receives equally the four segments types. Moreover, we also want to explore different loss functions: both sampling strategies have been evaluated using cross entropy and then a weighted cross entropy (Eq. 3.6). The four experiments were done using the Masked V-Net network [12].

In Table 4.3 we show the numerical results from the aforementioned experiments. Foreground-background sampling achieves the best performance among all in all tumor regions. Note that weighting the loss functions results in worse results than not using it.

Scheme	**Dice Whole**	**Dice Core**	**Dice Enhance**
Foreground-background sampling	**0,80156**	**0,59277**	**0,59949**
Per-Class sampling scheme	0,74112	0,53220	0,567712
Foreground-background sampling + weighted loss	0,72799	0,49146	0,58795
Per-Class sampling scheme+ weighted loss	0,30150	0,19794	0,47847

TABLE 4.3: Comparison of mean validation DSC metrics for fixed-rule sampling schemes

The results of the mentioned above experiments were different than we thought. We were expecting better results for the weighted cross-entropy. We will proceed to analyze these 4 cases in depth.

4.3.1.1 Loss function: cross-entropy

Figures 4.6 and 4.10 show in blue the training dice score curves and in red the validation dice score curves. Meanwhile, Tables 4.4 and 4.5 show the confusion matrices for both experiments. Per-class sampling achieves similar or even better results in training than foreground-background sampling. However, it has low generalization power, since there might be a large mismatch between the distributions of the training and the testing sets (where patches are obtained sampling uniformly from the MRI image) due to the alterations in the sampling training scheme. Ideally, we would do a sweep for different optimization and regularization hyper-parameters as it is known they affect deeply the model's performance, though is highly computationally and time demanding.

G \ P %	**0**	**1**	**2**	**3**
0	**0,99866**	0,00060	0,00065	0,00009
1	0,15597	**0,58136**	0,21434	0,04833
2	0,20463	0,13616	**0,64296**	0,01626
3	0,13979	0,05340	0,09242	**0,71439**

TABLE 4.4: Confusion matrix for baseline foreground-background (validation)

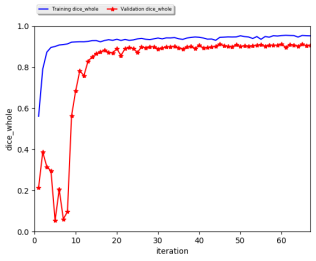


FIGURE 4.3: Dice Whole foreground-background

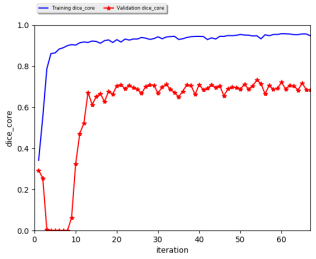


FIGURE 4.4: Dice Core foreground-background

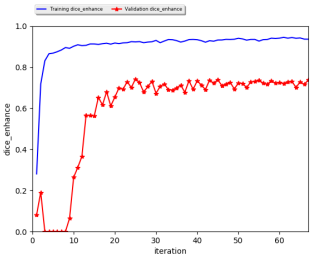


FIGURE 4.5: Dice Enhance foreground-background

FIGURE 4.6: Training / Validation Dice Score Evolution for baseline foreground-background

G \ P %	0	1	2	3
0	**0,99756**	0,00118	0,00113	0,00013
1	0,15377	**0,49406**	0,30911	0,04306
2	0,22491	0,08771	**0,67783**	0,00955
3	0,13086	0,04143	0,11687	**0,71083**

TABLE 4.5: Confusion matrix Per-Class sampling scheme (validation)

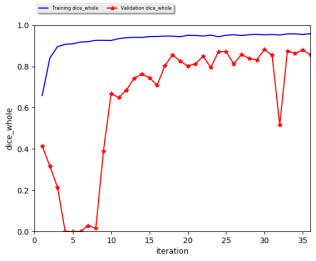


FIGURE 4.7: Dice Whole Per-Class sampling scheme

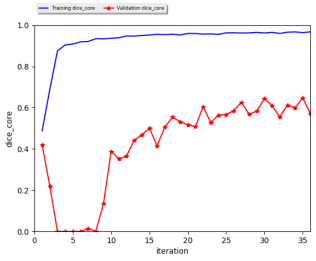


FIGURE 4.8: Dice Core Per-Class sampling scheme

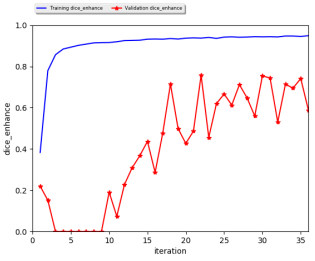


FIGURE 4.9: Dice Enhance Per-Class sampling scheme

FIGURE 4.10: Training / Validation Dice Score Evolution for Per-Class sampling scheme

4.3.1.2 Loss function: weighted cross-entropy

The choice of the weighted cross-entropy function double checks that jointly with the class-weighted sampling it influences the dice score of each class, defining the model's behavior, which leads to inefficient training.

Tables 4.6 and 4.7 and Figures 4.14 and 4.18 show the results obtained with this experiment. As it could be expected, in Table 4.6 we can see that the number of True-Positives for class 1 has increased but in return the number of True-Positives for class 2 has decreased and for almost all classes False-Positves have grown (in comparison to Table 4.4). In figure 4.10 it can be observed a bad performance for the combination of equiprobale-weighted class sampling and cross entropy weighted function. The corresponding confusion matrix (Table 4.7) only confirms this results.

G \ P %	0	1	2	3
0	**0,99723**	0,00213	0,00051	0,00013
1	0,13037	**0,62513**	0,18865	0,05585
2	0,22122	0,19958	**0,56491**	0,01429
3	0,12850	0,04857	0,08909	**0,73385**

TABLE 4.6: Confusion matrix for foreground-background sampling using weighted loss (validation)

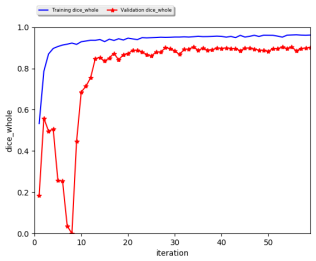


FIGURE 4.11: Dice Whole per foreground-background

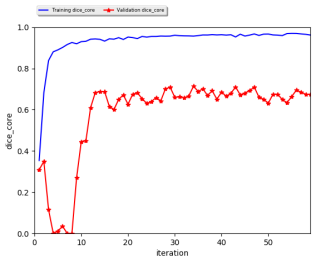


FIGURE 4.12: Dice Core per foreground-background

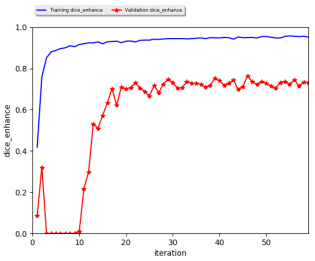


FIGURE 4.13: Dice Enhance per foreground-background

FIGURE 4.14: Training / Validation Dice Score Evolution for foreground-background sampling scheme and weighted loss

G \ P %	0	1	2	3
0	**0,84183**	0,15361	0,00364	0,00091
1	0,12418	**0,47787**	0,33199	0,06596
2	0,17954	0,09942	0,**70468**	0,01635
3	0,07395	0,02016	0,12268	**0,78322**

TABLE 4.7: Confusion matrix Per-Class sampling scheme using weighted loss (validation)

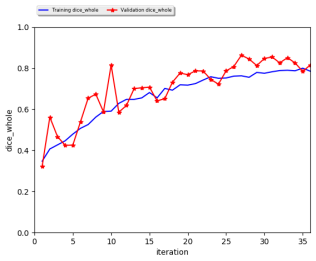


FIGURE 4.15: Dice Whole Per-Class sampling scheme

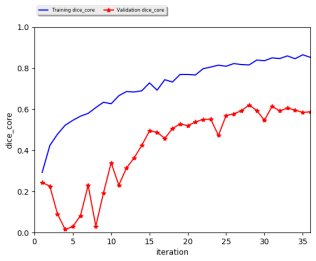


FIGURE 4.16: Dice Core Per-Class sampling scheme

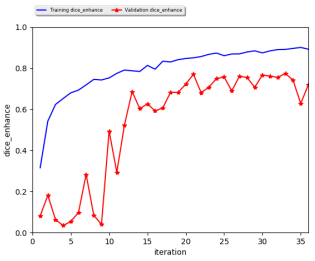


FIGURE 4.17: Dice Enhance Per-Class sampling scheme

FIGURE 4.18: Training / Validation Dice Score Evolution for Per-Class sampling scheme and weighted loss

4.3.2 Adaptive sampling schemes

Training with the fixed distribution is a simple approach in patch-based segmentation. Instead, it is possible to implement an adaptive sampling scheme according the result of the segmentation. Another way to achieve adaptive sampling consists in first, using the hardest patches to

discriminate, and then adding gradually the easiest ones. Consequently, the distribution on each iteration is different from the previous one.

4.3.2.1 CASED

The CASED method has been analyzed and studied to improve its performance in the tumor segmentation task. To make a fair comparison we employed Adam optimization [28] and masked V-Net network for all methods with the same fixed hyper-parameters. The learning rate was set to 0.0005. The loss chosen was cross entropy. Baseline CASED was implemented as explained in Chapter 3. Two generators were used according a curriculum (a rule to decide which generator use): one uniform generator and one only generating tumor patches. However, lesion patches were selected according the real relative tumor subregion distribution.

We compare the proposed CASED with two merging strategies: 1) We know that the distribution between the tumor sub-regions is not equiprobable. So, we decided to alter the generator that selects patches with tumor so that we chose patches of class 1,2,3 with the following probabilities: 40 %, 30 % and 30 % respectively. Instead of making it equiprobable (33,3 %), we decided to give a slightly higher weight to class 1 since it is the class that is more difficult to discriminate correctly; 2) Slowing down the curriculum curve with the K factor in order to delay the introduction of all type of patches (uniform sampling) and extend the number of epochs in which the network is trained with *difficult* patches. Variant 1 is also included in this method. More details are available in the Annex.

Table 4.8 presents the validation mean dice score for the three experiments carried with CASED scheme, while Table 4.9 and Table 4.10 show the corresponding confusion matrices illustrating the quality of the prediction and the the quantity of True-Positives vs False-Positives encountered in baseline CASED and in the slowed down version which include both models. The results from 4.10 are clearly better than 4.9. The main difference observed is the prediction of class 1 where it can be seen that thanks to adjusting the tumor subregion distribution and slowing down the curriculum curve the number of False-Positves has fallen down.

Scheme	**Dice Whole**	**Dice Core**	**Dice Enhance**
Baseline CASED	0,79089	0,62054	0,59778
Altered distribution CASED	0,83558	0,69727	0,63157
Slowed down CASED	**0,84130**	**0,73418**	**0,65403**

TABLE 4.8: Comparison of mean validation DSC metrics for CASED

G \ P %	**0**	**1**	**2**	**3**
0	**0,99937**	0,00005	0,00051	0,00006
1	0,19721	**0,35371**	0,35201	0,09707
2	0,24116	0,03398	**0,70236**	0,02250
3	0,11997	0,02711	0,08073	**0,77218**

TABLE 4.9: Confusion matrix baseline CASED scheme (validation)

Figure 4.19 and figure 4.20 show the evolution of the number of patches of any type and the number of patches with tumor according to each training epoch. For clarification, we are only

G \ P %	0	1	2	3
0	**0,99877**	0,00016	0,00095	0,00012
1	0,10293	**0,57165**	0,24711	0,07831
2	0,15830	0,04181	**0,78414**	0,01576
3	0,05627	0,04030	0,06982	**0,83361**

TABLE 4.10: Confusion matrix slowed down CASED scheme (validation)

taking into account the central voxel to decide which patch type is it. If we considered all the patch, calculating the class with maximum presence we would always obtain class 0. The third plot is the curriculum curve from equation 3.8. Overall, the slowed down version was found to have the best performance.

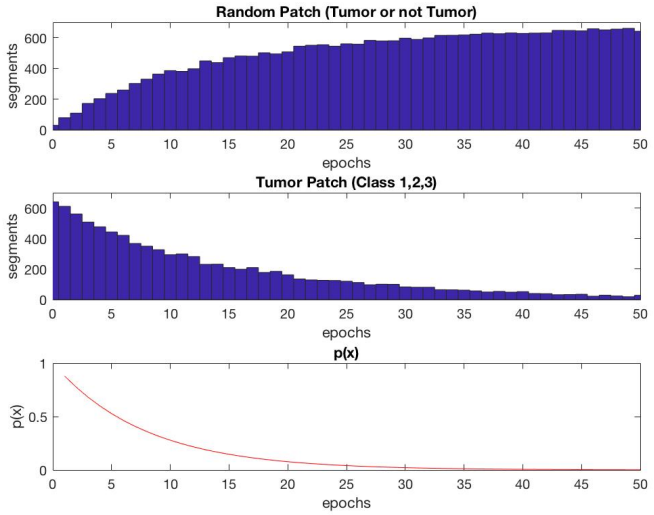


FIGURE 4.19: Patch distribution evolution during training in baseline CASED model

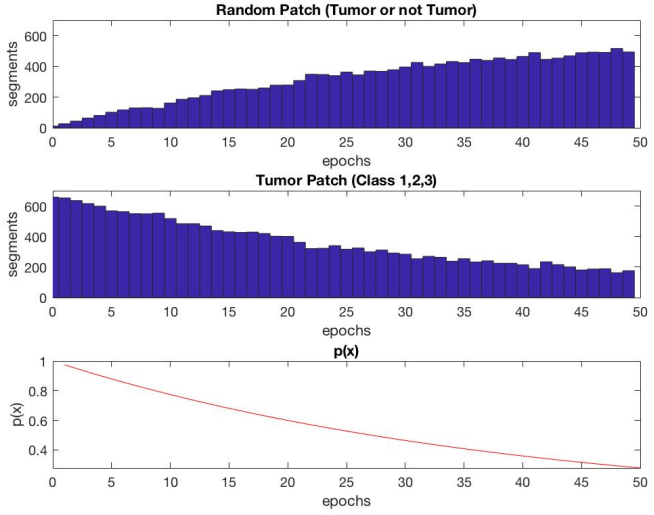


FIGURE 4.20: Patch distribution evolution during training in slowed down model

Finally, the model was used to infer the test data segmentation. Boxplots from figure 4.24 show the comparison in DSC across the three models. The outliers had been checked and were mainly due to MRI input images from the dataset with poor conditions. In conclusion, variant 2 demonstrates to be better than baseline CASED or only including variant 1 according to the mean DSC metrics, the confusion matrix and the validation boxplots. Until this point, this result outperforms all the other schemes used previously. This is because in the last iterations the training distribution approaches the true distribution and therefore gets a better generalization.

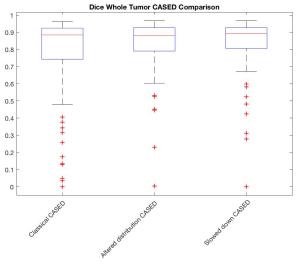


FIGURE 4.21: Validation Dice Whole CASED

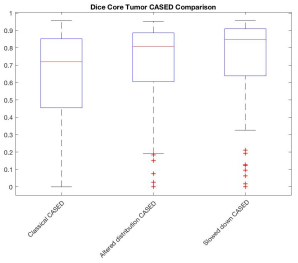


FIGURE 4.22: Validation Dice Core CASED

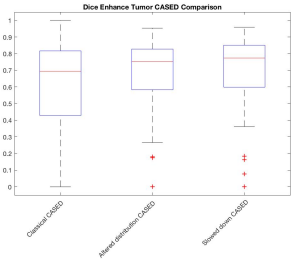


FIGURE 4.23: Validation Dice Enhance CASED

FIGURE 4.24: Test DSC for all three CASED variants. From left to right: Baseline CASED, modified subregion distribution and slowed down CASED. Validation is done using the whole subject as input.

4.3.2.2 BaseASS

The baseline adaptive sampling scheme was implemented as mentioned in Chapter 3. The baseline method uses cross-entropy and the central voxel error as the criteria of selection. Moreover, we explore this model and propose three alternatives: 1) patch selection according the median error value of the whole patch (Eq. 4.1) instead of only the error value of the central voxel (Eq. 3.10); 2) use the Generalised DSC loss function instead of cross-entropy; 3) use another architecture: Deep Medic network instead of masked V-Net. For this last experiment, the cost function chosen was cross-entropy as it had shown to perform better in this configuration. This classical architecture was chosen because, unlike Masked V-Net [12], it does not have any max-pooling layer. We want to avoid max-pooling beacuse it reduces the spatial size of the representation and thereby it reduces the number of training parameters, which might contain relevant information for our segmentation task.

$$median(E_{i_{patch}}) > U(0,1) - \epsilon \tag{4.1}$$

To be as fair as possible when comparing, the rest of parameters remained fixed. We trained baseline and the first two models with Adam Optimization method [28] with a learning rate of 0.0005. The third model (Deep Medic Network) was trained using RMSprop optimizer and a learning rate of 0.0001. For more details consult the annex.

The results reported on Table 4.11 show that BaseASS with classical cross-entropy loss outperforms Generalised DSC. However, BaseASS using the central voxel's error or the median error

Scheme	Dice Whole	Dice Core	Dice Enhance
BaseASS	**0,86286**	**0,74418**	**0,67466**
BaseASS + Generalised DSC	0,81759	0,69727	0,65100
BaseASS + Median error as selection criteria	0,85087	0,75052	0,657198
BaseASS + Deep Medic network	0,65379	0,55027	0,48243

TABLE 4.11: Comparison of mean validation DSC metrics for BaseASS

of the patch show similar performance. This suggests that using only the central voxel is enough to get a general representation of the patch error. This results can be clearly confirmed when looking at the boxplot comparison in Figure 4.28. Finally, the masked V-Net network obtains better results than the Deep Medic Network. Nevertheless, the latter hyper-parameters had not been optimized and we can't conclude that it is behaving worse yet. The last experiment was not worth to be included in the boxplot comparison 4.28, but can be found in the Appendix in Figure C.25.

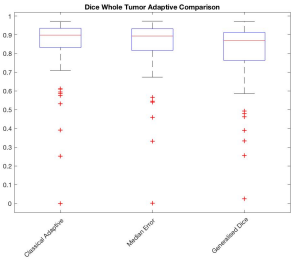


FIGURE 4.25: Validation Dice Whole BaseASS

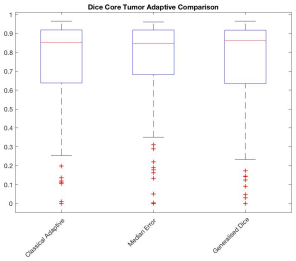


FIGURE 4.26: Validation Dice Core BaseASS

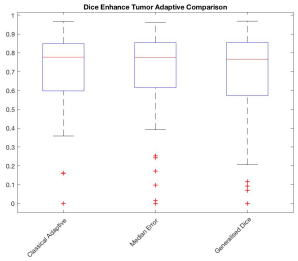


FIGURE 4.27: Validation Dice Enhance BaseASS

FIGURE 4.28: Test DSC for all three BaseASS variants. From left to right: BaseASS using cross entropy loss, BaseASS using median error and BaseASS using generalised DSC loss. Validation is done using the whole subject as input

Figure 4.29 shows the number of segments of each class per training epoch for the BaseASS model. For clarification, it is considered that the patch class is defined as the class of the central voxel (the one whose error value is taken as selection criteria). In comparison to CASED scheme, section 4.3.2.1 (Figure 4.19), here the number of lesion patches is not reduced in each epoch but remains constant for all the training. In contrast, the number of background patches decreases in each iteration. It can be seen that the portion of class 1 patches is slightly higher than that of class 2 and 3. This leads us to believe that it is a wise decision to give a little more weight to class 1 than 2 or 3.

While Figure 4.30 shows the training distribution of the batch in each iteration. It can be observed that despite of choosing different number of segments of each class, the training distribution remains almost constant for all the epochs. Moreover, this distribution is close to the real one.

Qualitative results of the error maps calculated in each iteration (Figure 4.31) demonstrate the importance of selecting the right lesion patches (those with high error values) against selecting any tumor patch without any criteria. It can be seen how the performance of the networks

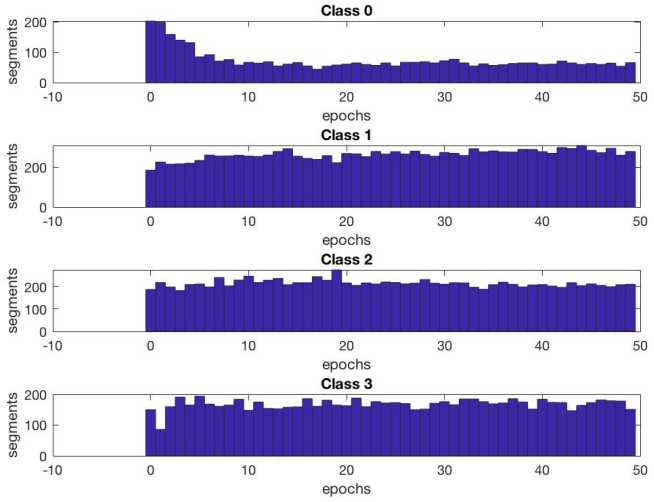


FIGURE 4.29: Patch distribution evolution for the BaseASS model

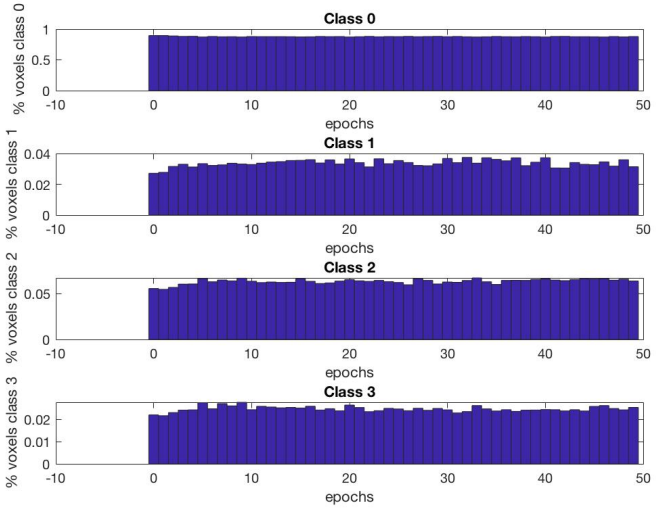
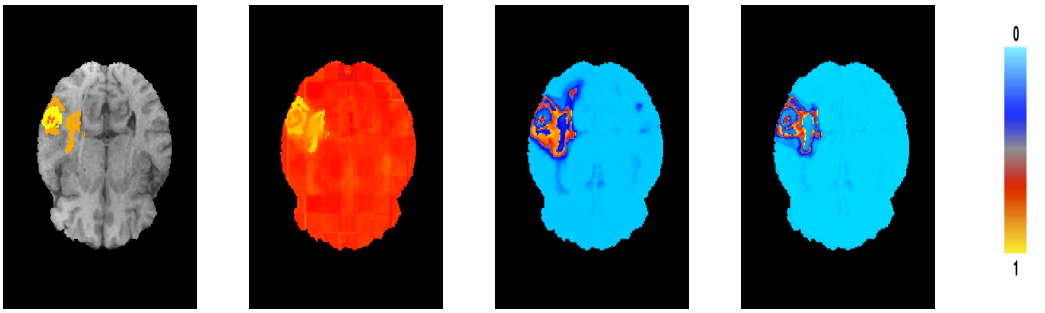


FIGURE 4.30: Percentage of class voxels per batch per epoch for BaseASS

improves from epoch 1 (Fig. 4.31(b)) to epoch 30 (Fig. 4.31(d)). We can see that from the 15th to the 30th epoch, the network works to refine the segmentation. Mainly, the error lies in the boundaries between two different classes.

Table 4.12, Table 4.13 and Table 4.14 show the confusion matrices for the BaseASS with the two different loss functions and different architecture.



(a) Ground Truth above MRI T1 (b) Error map in epoch 1 (c) Error map in epoch 15 (d) Error map in epoch 30 (e) Color scale

FIGURE 4.31: Error maps from subject $Brats17-CBICA-AAL$ with BaseASS. Light blue is equivalent to zero error, yellow means maximum error value.

G \ P %	0	1	2	3
0	**0,99914**	0,00013	0,00066	0,00007
1	0,09024	**0,61238**	0,22317	0,07421
2	0,15804	0,06343	**0,76245**	0,01608
3	0,05271	0,04601	0,06876	**0,83252**

TABLE 4.12: Confusion matrix BaseASS with cross-entropy loss (validation)

G \ P %	0	1	2	3
0	**0,99825**	0,00020	0,00140	0,00015
1	0,11032	**0,62370**	0,18580	0,08018
2	0,13554	0,062621	**0,78282**	0,01903
3	0,05122	0,04287	0,06184	**0,84407**

TABLE 4.13: Confusion matrix BaseASS with Generalised DSC loss (validation)

G \ P %	0	1	2	3
0	**0,99361**	0,00074	0,00469	0,00096
1	0,12832	**0,52653**	0,29216	0,05300
2	0,11627	0,09170	**0,76764**	0,02439
3	0,04074	0,13621	0,07489	**0,74816**

TABLE 4.14: Confusion matrix for BaseASS and Deep Medic network (validation)

4.4 Discussion

In this work, we have analyzed and evaluated 13 sampling schemes. An overall comparison is done in Table 4.15. First, we concluded that for whole subject (dense-training) DSC performed better than cross-entropy since the second treats all training voxels equally and this is not helpful when the network has difficulties in learning representations of the minority classes. DSC does an implicit re-weighting of the voxels alleviating this issue.

In the context of patch sampling, fixed-rule sampling schemes have appeared to be risky models as they over-modifiy the training distribution from the original one, in such manner that they change the model behavior making it difficult to adjust to the validation distribution (uniform).

The best result was for foreground-background sampling without weighted loss function, the scheme among the four which less modifies the distribution. This results have been checked for cross-entropy, but we cannot ensure what would happen using other loss functions.

Regarding adaptive sampling, both CASED [25] and BaseASS [26] achieve very good results. It's risky to claim which one is better than the other because, although the BaseASS presents the higher dice score and even a little more better results on the confusion matrix, the behavior is very similar and we do not know if we could achieve better results adjusting the models with new modifications. The key to the success of both is that the final distribution at the training end is very close to the real one. Hence, this gives them strength to alleviate class imbalance and generalize correctly.

Scheme	**Dice Whole**	**Dice Core**	**Dice Enhance**
Dense-training + Cross-entroy	0,67032	0,41302	0,50329
Dense-training + DSC	0,80534	0,70092	0,65296
Foreground-background sampling scheme	0,80156	0,59277	0,59949
Per-Class sampling scheme	0,74112	0,53220	0,567712
Foreground-background sampling scheme + WL	0,72799	0,49146	0,58795
Per-Class sampling scheme+ WL	0,30150	0,19794	0,47847
CASED	0,84130	0,73418	0,65403
BaseASS	**0,86286**	**0,74418**	**0,67466**

TABLE 4.15: Mean Dice score metrics from the main experiments carried on

Boxplots 4.32, 4.33 and 4.34 show quartile ranges of the DSC scores (Whole Tumor, Core Tumor and Enhancing Tumor respectively) on the test datasets, dots indicate outliers and the red line indicates the median value. They give us an overview of the behaviour of each of the experiments. It could be argued that whole subject with DSC has a comparable outcome to BaseASS or CASED, however, if we look at quartile 25 of the first scheme, it is much lower than the quartile 25 of the last two.

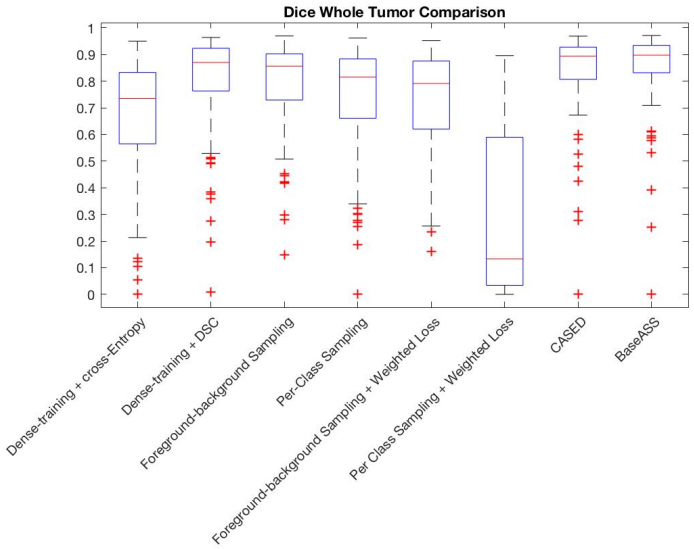


FIGURE 4.32: Overall Dice Whole Tumor Comparison. Validation is done using the whole subject as input.

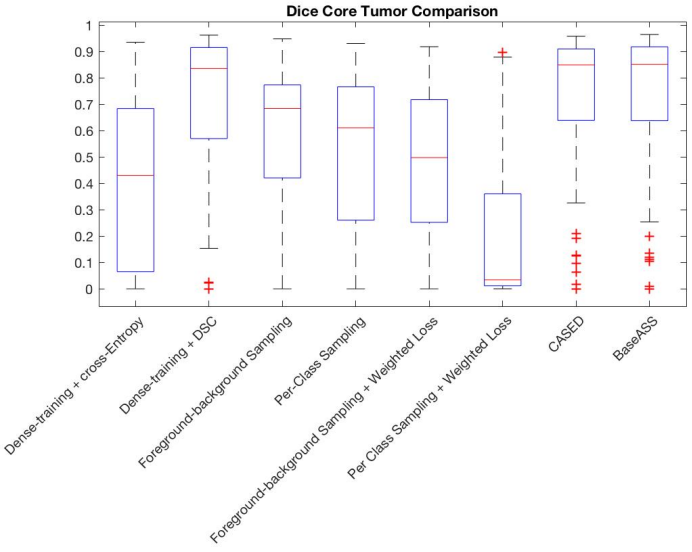


FIGURE 4.33: Overall Dice Core Tumor Comparison. Validation is done using the whole subject as input.

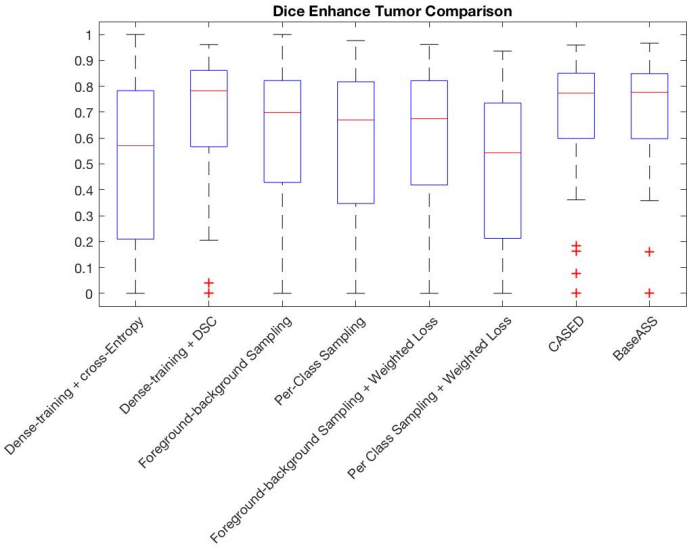


FIGURE 4.34: Overall Dice Enhance Tumor Comparison. Validation is done using the whole subject as input.

Chapter 5

Budget

This project has been carried in the Image Processing Group, ETSETB, UPC. Deep Learning is highly computationally demanding, consequently a GPU was needed: The GPU GeForce GTX Titan Black has an approximate cost of 920 €, however UPC provided it to us without any cost.

Thus, the main cost of this project comes from the salary of the researchers and the time spent in it. The team for the development of this thesis is formed by two professors who were advising me as senior engineers and myself as junior engineer. The total duration of the project was of 33 weeks. The budget of the project can be calculated:

	Amount	**Wage/hour**	**Dedication**	**Total**
Junior Engineer	1	8,00€/h	25h/week	6,600 €
Senior Engineer	2	20,00€/h	4h/week	5,280 €

Table 5.1: Project Budget

Chapter 6

Conclusions and future development

The main goal of this project was to apply different state-of-the art methodologies to brain tumor segmentation to make a comparative study of all them. We studied how dense-training and patch sampling performed in the brain tumor segmentation. For this reason, we proposed variants to fixed-rule and adaptive sampling schemes.

In section 4.2 we proved that the DSC outperforms cross-entropy ability to classify tumor/non-tumor voxels in dense-training. However, in patch sampling, cross-entropy demonstrated to be better than weighted loss functions as weighted cross-entropy and generalised DSC.

Then, in section 4.3, we show that properly designed patch sampling outperforms dense-training schemes. Moreover, we conclude that if we alter significantly the training distribution from the real, such that using per-class sampling scheme, we increase generalization error even if training improves due to the mismatch between training and testing distributions. Moreover, novel adaptive training schemes are shown to further improve the performance compared to the fixed-rule schemes for the brain tumor segmentation task. We observed how only adaptive sampling obtaines good results altering the training distribution in such a way that achieves learning the features of those segments that are more difficult to classify. Our best results are: dice score for whole tumor of 0,862 , dice score for core tumor of 0,744 and a dice score for enhancing tumor of 0,67.

We compare our results with the leaders of the MICCAI challenge ranking and the results presented by the UPC this last edition (Table 6.1). For dice score of whole tumor, our method is very close to the results obtained by the top performing methods while, our method achieves low dice score for enhancing tumor and core tumor. BaseASS has similar performance to the approach presented by the UPC, in which they tackle the problem with a pipeline of two masked V-Nets and dense-training. However, our results are still well below the ensemble model proposed by UCL-TIG.

Scheme	**Dice Whole**	**Dice Core**	**Dice Enhance**
UCL-TIG	0,9	0,83	0,78
UPC	0,87	0,63	0,71
BaseASS	**0,86286**	**0,74418**	**0,67466**

TABLE 6.1: Comparison between our approach and some of 2017 MICCAI BraTS Challenge

Finally, the choice of different hyper-parameters for the optimization and regularization can heavily affect the performance of a model. It is often observed that the choice of optimizer and its configuration, for instance regularitzation or the learning rate, to a large extent, determine whether a good or a bad segmentation is obtained. The sensitivity to all these hyper-parameters is magnified by the fact that re-using the same setting does not guarantee to behave well among different network architectures, or even on different tasks and data. Hence, it is often difficult to draw generic and confident conclusions without spending a huge amount of time in optimizing the experimental settings.

In the future, we are interested in trying training with a different architectures as we belief that this one might be a bottleneck. We are interested in trying dilated convolutions as they are able to introduce systematically multi-scale contextual information without loosing resolution. HighResNet [16] is a network which uses dilated convolution and residual connections. The architecture is based on the fact that dilated convolutions support exponential expansion of the receptive field without loss of resolution and avoid max-pooling.

Now that we already know that BaseASS provides sucessful results it would be interesting to do more reserach on it. The BaseASS and also the CASED still have other learning parameters that could be explored, for instance give priority to patches of the modality with the greatest impact. It would also be relevant for consideration make adaptive other parameters as the learning rate or the patch-size. It also has been left trying to do an ensemble of the best methods in this thesis. Finally, adapative sampling schemes from this work might be powerful in other segmentation tasks where imbalance is also a problem, like white matter hyperintensities (WMH) segmentation.

Appendix A

Code of the project

The code of the project can be found in GitHub repository [29]. It has been fully developed in python using Keras with Tensorflow backend.

Appendix B

Dense-training

B.1 Set up

The set up used for the mentioned experiment:

- Network: Masked V-Net
- Optimization: Adam
- No Data augmentation
- Regularization: l1=0.00001, l2=0.005
- Momentum: 0.99
- Learning Rate: 0.0005

B.2 Training Curves

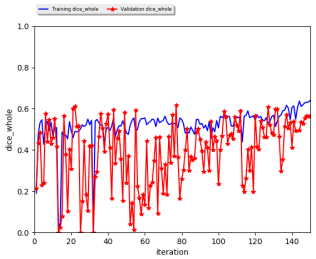


FIGURE B.1: Dice Whole

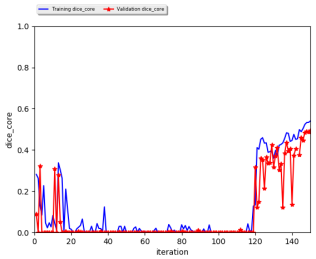


FIGURE B.2: Dice Core

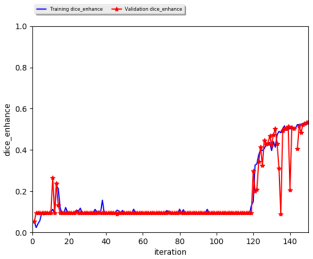


FIGURE B.3: Dice Enhance

FIGURE B.4: Dice Score Evolution for cross-entropy loss function

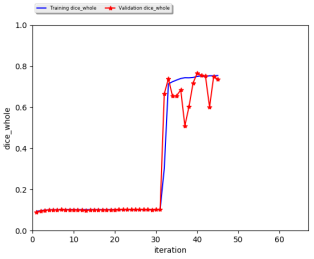


FIGURE B.5: Dice Whole

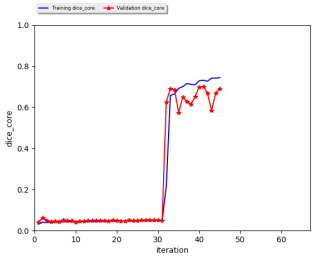


FIGURE B.6: Dice Core

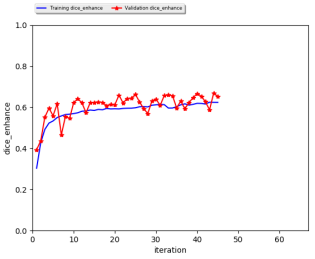


FIGURE B.7: Dice Enhance

FIGURE B.8: Dice Score Evolution for DSC loss function

Appendix C

Patch sampling

C.1 Set up

To perform a correct analysis, the same set up was used for all patch sampling experiments:

- Loss: cross-entropy / weighted cross-entropy/ Generalised DSC
- Network: Masked V-Net / Deep Medic Network
- Optimization: Adam / RMSprop
- Epochs: 50
- Segments Train / epoch: 600
- Segments Validation / epoch: 400
- Data augmentation: False
- Regularization: l1=0.00001, l2=0.005
- Momentum: 0.99
- Learning Rate: 0.0005 / 0.0001

C.2 Training Curves

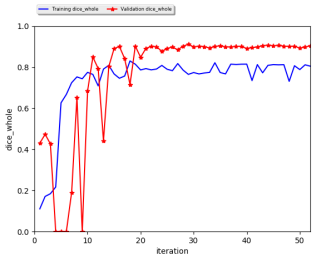


FIGURE C.1: Dice Whole

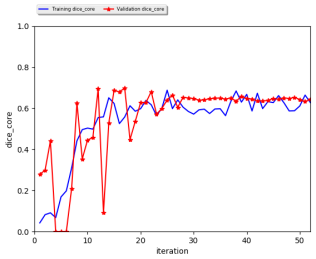


FIGURE C.2: Dice Core

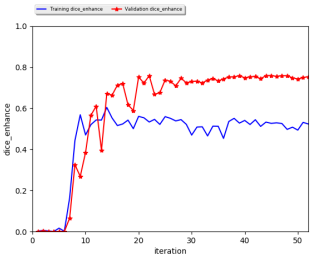


FIGURE C.3: Dice Enhance

FIGURE C.4: Dice Score Evolution for the baseline CASED model

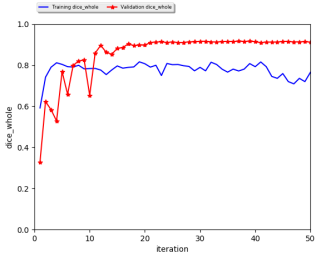


FIGURE C.5: Dice Whole

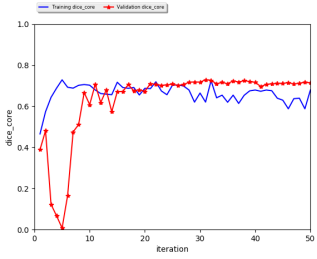


FIGURE C.6: Dice Core

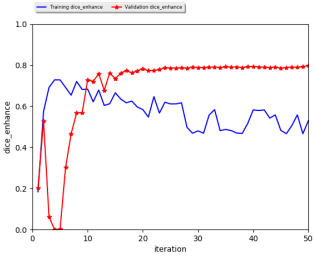


FIGURE C.7: Dice Enhance

FIGURE C.8: Dice Score Evolution for the altered distribution CASED model

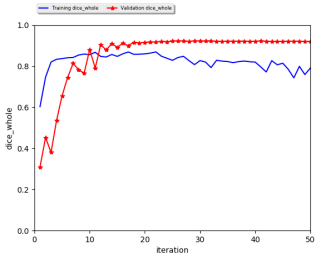


FIGURE C.9: Dice Whole

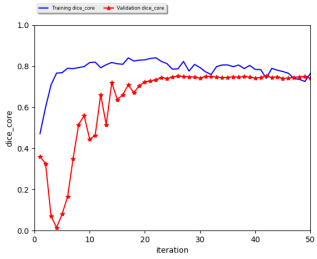


FIGURE C.10: Dice Core

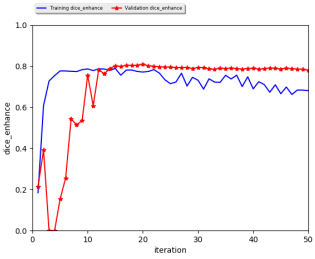


FIGURE C.11: Dice Enhance

FIGURE C.12: Dice Score Evolution for the slowed down distribution CASED model

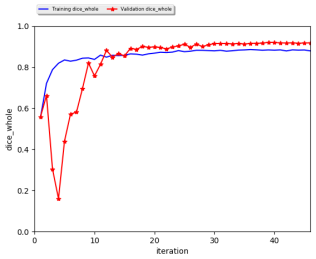


FIGURE C.13: Dice Whole

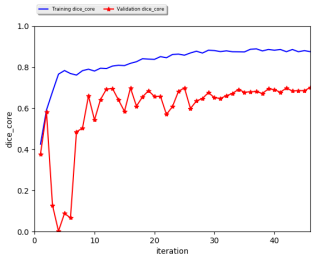


FIGURE C.14: Dice Core

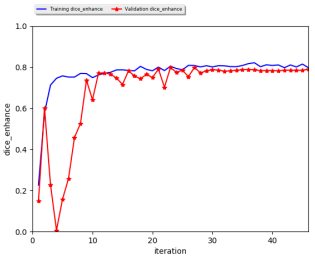


FIGURE C.15: Dice Enhance

FIGURE C.16: Dice Score Evolution for BaseASS model

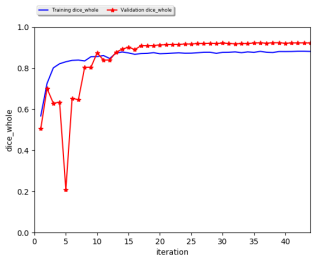


FIGURE C.17: Dice Whole

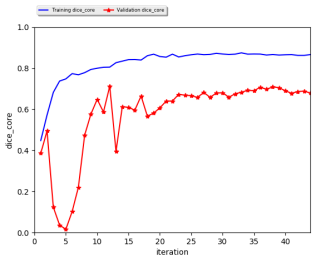


FIGURE C.18: Dice Core

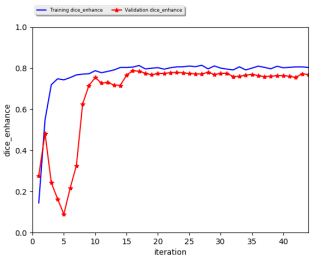


FIGURE C.19: Dice Enhance

FIGURE C.20: Dice Score Evolution for BaseASS model and median error

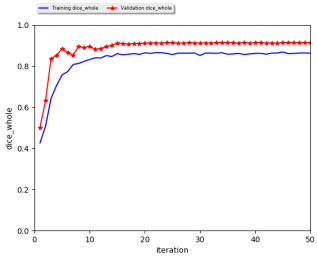


FIGURE C.21: Dice Whole

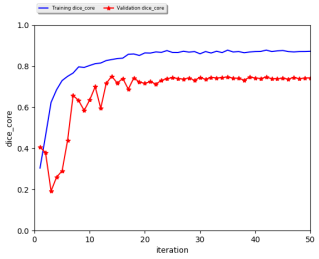


FIGURE C.22: Dice Core

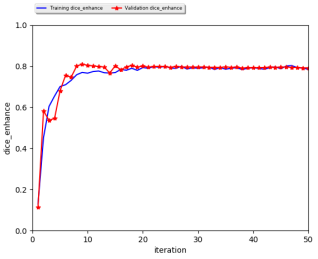


FIGURE C.23: Dice Enhance

FIGURE C.24: Dice Score Evolution for BaseASS and generalised DSC

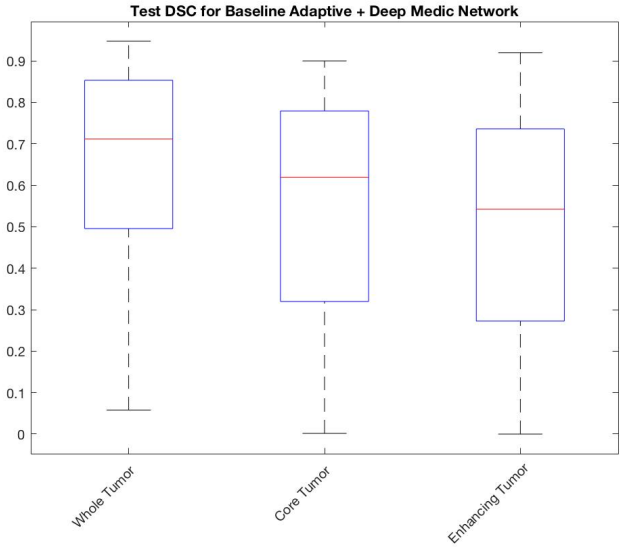


FIGURE C.25: Test DSC for BaseASS and Deep Medic network

Bibliography

[1] E.C. Holland. Progenitor cells and glioma formation.

[2] Bjoern Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elisabeth Gerstner, Marc-Andre Weber, Tal Arbel, Brian Avants, Nicholas Ayache, Patricia Buendia, Louis Collins, Nicolas Cordier, Jason Corso, Antonio Criminisi, Tilak Das, Hervé Delingette, Cagatay Demiralp, Christopher Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andac Hamamci, Khan Iftekharuddin, Raj Jena, Nigel John, Ender Konukoglu, Danial Lashkari, Jose Antonio Mariz, Raphael Meier, Sergio Pereira, Doina Precup, S. J. Price, Tammy Riklin-Raviv, Syed Reza, Michael Ryan, Lawrence Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos Silva, Nuno Sousa, Nagesh Subbanna, Gabor Szekely, Thomas Taylor, Owen Thomas, Nicholas Tustison, Gozde Unal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koen Van Leemput. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, October 2014. doi: 10.1109/TMI.2014.2377694. URL https://hal.inria.fr/hal-00935640.

[3] Multimodal brain tumor segmentation challenge 2017. URL http://www.med.upenn.edu/sbia/brats2017/data.html.

[4] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, . Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, Oct 2015. ISSN 0278-0062. doi: 10.1109/TMI.2014.2377694.

[5] Fei-Fei Li, Justin Johnson, and Serena Yeung. Cs231n: Convolutional neural networks for visual recognition., 2017. URL http://cs231n.stanford.edu/.

[6] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron C. Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *CoRR*, abs/1505.03540, 2015. URL http://arxiv.org/abs/1505.03540.

[7] et al Adrià Casamitjana. 3D Convolutional Neural Networks for Brain Tumor Segmentation: a comparison of multi-resolution architectures. *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. Springer, Cham, 2016.*

[8] François Chollet et al. Keras. https://keras.io, 2015.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[10] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A. Silva. Deep convolutional neural networks for the segmentation of gliomas in multi-sequence mri. In Alessandro Crimi, Bjoern Menze, Oskar Maier, Mauricio Reyes, and Heinz Handels, editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 131–143, Cham, 2016. Springer International Publishing. ISBN 978-3-319-30858-6.

[11] Konstantinos Kamnitsas, Christian Ledig, Virginia F. J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d CNN with fully connected CRF for accurate brain lesion segmentation. *CoRR*, abs/1603.05959, 2016. URL http://arxiv.org/abs/1603.05959.

[12] Marcel Catà, Adrià Casamitjana, Irina Sánchez, Marc Combalia, and Verónica Vilaplana. Masked v-net: an approach to brain tumor segmentation.

[13] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *CoRR*, abs/1606.06650, 2016. URL http://arxiv.org/abs/1606.06650.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

[15] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015. URL http://arxiv.org/abs/1505.00387.

[16] Wenqi Li, Guotai Wang, Lucas Fidon, Sébastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. On the compactness, efficiency, and representation of 3d convolutional networks: Brain parcellation as a pretext task. *CoRR*, abs/1707.01992, 2017. URL http://arxiv.org/abs/1707.01992.

[17] *Preproceedings 2017 International MICCAI BraTS Challenge*, 2017. MICCAI,CBICA. URL https://www.cbica.upenn.edu/sbia/Spyridon.Bakas/MICCAI_BraTS/MICCAI_BraTS_2017_proceedings_shortPapers.pdf.

[18] Konstantinos Kamnitsas, Wenjia Bai, Enzo Ferrante, Steven G. McDonagh, Matthew Sinclair, Nick Pawlowski, Martin Rajchl, Matthew C. H. Lee, Bernhard Kainz, Daniel Rueckert,

and Ben Glocker. Ensembles of multiple models and architectures for robust brain tumour segmentation. *CoRR*, abs/1711.01468, 2017. URL http://arxiv.org/abs/1711.01468.

[19] Tom Brosch, Youngjin Yoo, Lisa Y. W. Tang, David K. B. Li, Anthony Traboulsee, and Roger Tam. Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 3–11, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[20] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *CoRR*, abs/1606.04797, 2016. URL http://arxiv.org/abs/1606.04797.

[21] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *CoRR*, abs/1707.03237, 2017. URL http://arxiv.org/abs/1707.03237.

[22] *Multimodal Brain Tumor Image Segmentation Benchmark: Change Detection*, 2016. MICCAI,CBICA.

[23] Cs229: Additional notes on backpropagation, 2017. URL http://cs229.stanford.edu/notes/cs229-notes-backprop.pdf.

[24] Lucas Fidon, Wenqi Li, Luis C. García-Peraza-Herrera, Jinendra Ekanayake, Neil Kitchen, Sébastien Ourselin, and Tom Vercauteren. Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. *CoRR*, abs/1707.00478, 2017. URL http://arxiv.org/abs/1707.00478.

[25] Jesson A., Guizard N., Ghalehjegh S.H., Goblot D., Soudan F., and Chapados N. Cased: Curriculum adaptive sampling for extreme data imbalance. 10435, 2017.

[26] Lorenz Berger, Eoin Hyde, M. Jorge Cardoso, and Sébastien Ourselin. An adaptive sampling scheme to efficiently train fully convolutional networks for semantic segmentation. *CoRR*, abs/1709.02764, 2017. URL http://arxiv.org/abs/1709.02764.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. URL http://arxiv.org/abs/1502.01852.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

[29] Thesis. URL https://github.com/clarabonnin/segmentation_DLMI_clara.